

II. AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently Amended) A language extension to an object oriented programming language for the allocation of resource objects to users in a programming environment, the language extension comprising:

a resource pool for storing a plurality of individual available resource objects;

a resource queue for storing data representing select ones of the users, the users on the resource queue seeking the allocation of the resource objects;

a release resource system for releasing a concluded resource object after use by a user, comprising:

means for returning the concluded resource object to the resource pool; and

means for, when the resource queue is not empty, using an object oriented programming language notify method to notify the users waiting on a synchronization object that the concluded resource object is available;

an acquire resource system for an acquiring user to acquire at least one of the resource objects, comprising:

means for returning at least one of the resource objects in the resource pool if the resource pool is not empty and there are no users on the resource queue;

when the resource pool is empty or the resource queue is non-empty:

means for adding the acquiring user to the resource queue;

means for having the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object;

means for awakening the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object;

means for awakening the acquiring user when there is a resource object in the resource pool;

means for determining if the acquiring user is at the head of the resource queue, and if the acquiring user is at the head of the resource queue:

means for removing the acquiring user from the resource queue;

means for removing at least one of the resource objects from the resource pool; and

means for returning said at least one of the resource objects to the acquiring user, otherwise, continuing to have the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object; and

means for the acquire resource system to ~~return a time out exception~~ remove the acquiring user from the resource queue if the acquiring user does not acquire said at least one of the resource objects within a predefined time; and

synchronization means to constrain users with regard to a plurality of individual available resource objects such that only one user may execute either the release resource system or the acquire resource system at any one time, said synchronization means being synchronized on the synchronization object.

2. (Original) The language extension of Claim 1, in which the synchronization object is the resource queue.

3. (Previously Presented) The language extension of claim 2, in which the programming language is Java.

4. (Previously Presented) The language extension of claim 3 in which the acquire resource system comprises a helper means to wait for the acquiring user to be at the head of the resource queue and the resource pool to be non-empty.

5. (Currently Amended) A method for allocating resource objects to users in a programming language environment, the programming language environment comprising a resource pool for storing a plurality of individual available resource objects, and a resource queue for storing data representing select ones of the users seeking allocation of the resource objects, the method comprising the steps of:

- a. releasing a concluded resource object after use by a user, comprising the steps of:
 - i. returning the concluded resource object to the resource pool; and
 - ii. when the resource queue is not empty, using an object oriented programming language notify method to notify users waiting on a synchronization object that the concluded resource object is available.
- b. permitting an acquiring user to acquire at least one of the resource objects,

comprising the steps of:

- i. returning to the acquiring user at least one of the resource objects in the resource pool if the resource pool is not empty and there are no users on the resource queue;
 - ii. where the resource pool is empty or the resource queue is non-empty, adding the acquiring user to the resource queue, having the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object, awakening the acquiring user when there is a resource object in the resource pool, determining if the acquiring user is at the head of the resource queue, and if the acquiring user is at the head of the resource queue, removing the acquiring user from the resource queue, removing at least one of the resource objects from the resource pool, and returning said at least one of the resource objects to the acquiring user, otherwise, continuing to have the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object; and
 - iii. removing the acquiring user from the resource queue ~~returning a time out exception~~ if the acquiring user does not acquire at least one of the resource objects within a predefined time; and
- c. synchronizing each of the above steps to constrain users such that only one user with regard to a plurality of individual available resource objects may execute either the above steps (a) and (b) at any one time, said synchronization being

carried out on the synchronization object.

6. (Original) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps of Claim 5.

7. (Currently Amended) A program storage device readable by a machine, tangibly embodying instructions executable by the machine to provide a programming language extension to an object oriented programming language for the allocation of resource objects to users in a programming environment, the extension comprising:

a resource pool for storing a plurality of individual available resource objects;

a resource queue for storing data representing select ones of the users seeking allocation of the resource objects;

a release resource system for releasing a concluded resource object after use by a user, the release resource method comprising:

means for returning the concluded resource object to the resource pool;

means for, when the resource queue is not empty, using an object oriented programming language notify method to notify the users waiting on a synchronization object that the concluded resource object is available;

an acquire resource system for an acquiring user to acquire at least one of the resource objects, comprising:

means for returning at least one of the resource objects in the resource pool if the resource pool is not empty and there are no users on the resource queue;

when the resource pool is empty of the resource queue is non-empty:

means for adding the acquiring user to the resource queue;

means for having the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object;

means for awakening the acquiring user when there is a resource object in the resource pool;

means for determining if the acquiring user is a the head of the resource queue, and if the acquiring user is a the head of the resource queue:

means for removing the acquiring user from the resource queue;

means for removing at least one of the resource objects from the resource pool; and

means for returning said one of the resource object to the acquiring user, otherwise, continuing to have the acquiring user wait for notification by using the object oriented programming language wait method on the synchronized object; and

means for the acquire resource system to ~~return a time out exception~~ remove the acquiring user from the resource queue if the acquiring user does not acquire at least one of the resource objects within a predefined time; and

synchronization means to constrain users with regard to a plurality of individual available resource objects such that only one user may execute either the release resource system or the acquire resource method at any one time, said synchronization means being synchronized on the synchronization object.

8. (Original) The program storage device of Claim 7, in which the synchronization object is the

resource queue.

9. (Previously Presented) The program storage device of claim 8, in which the programming language is Java.

10. (Previously Presented) The program storage device of claim 9, in which the acquire resource system comprises a helper means to wait for the acquiring user to be at the head of the resource queue and the resource pool to be non-empty.

11. (Currently Amended) A computer program product for use with a computer supporting the Java programming language environment, the computer program product comprising a Java class for the allocation of resource objects to threads in a programming environment, the class comprising:

a resource pool for storing a plurality of individual available resource objects;

a resource queue for storing data representing select ones of the threads, the threads on the resource queue seeking the allocation of the resource objects;

a release resource system for releasing a concluded resource object after use by a thread, the release resource system comprising:

means for returning the concluded resource object to the resource pool;

when the resource queue is not empty, means for notifying the threads waiting on

a synchronization object that the concluded resource object is available;

an acquire resource system for an acquiring thread to acquire at least one of the resource

objects, the acquire resource system comprising:

means for returning at least one of the resource objects in the resource pool if the resource pool is not empty and there are no threads on the resource queue;

when the resource pool is empty or the resource queue is non-empty:

means for adding the acquiring thread to the resource queue;

means for invoking the Java wait method on the synchronized object;

when the acquiring thread is awakened and there is a resource object in the resource pool, means for determining if the acquiring thread is at the head of the resource queue, and if the acquiring thread is at the head of the resource queue:

means for removing the acquiring thread from the resource queue;

means for removing at least one of the resource objects from the resource pool; and

means for returning said at least one of the resource objects to the acquiring user, otherwise, invoking the Java wait method on the synchronized object; and

means for ~~returning a time out exception~~ removing the acquiring thread from the resource queue if the acquiring thread does not acquire at least one of the resource object within a predefined time; and

means for using the Java synchronized method to constrain threads with regard to a plurality of individual available resource objects such that only one thread may execute either the release resource system or the acquire resource system at any one time, said Java synchronized method being synchronized on the synchronization object.

12. (Object) The computer program product of Claim 11 in which the synchronization object is the resource queue.

13. (Previously Presented) The computer program product of claim 12, in which the acquire resource system comprises helper means to wait for the acquiring user to be at the head of the resource queue and the resource pool to be non-empty.

14. (Currently Amended) A language extension to a programming language for the allocation of resources to users in a programming environment, the language extension comprising:

a resource pool for storing a plurality of individual available resources;

a resource queue for storing data representing select ones of the users seeking allocation of the resource objects;

a release resource means for releasing a concluded resource after use by a user, the release resource means comprising:

means for returning the concluded resource to the resource pool; and

means for, when the resource queue is not empty, notifying the user waiting on a predefined synchronization flag that the concluded resource is available;

an acquire resource means for an acquiring user to acquire at least one of the resources, the acquire resource means comprising:

means for providing the acquiring user with at least one of the resources in the resource pool if the resource pool is not empty and there are no users on the resource

queue;

when the resource pool is empty or the resource queue is non-empty:

means for adding the acquiring user to the resource queue;

means for having the acquiring user wait for notification on the synchronization flag;

means for awakening the acquiring user when the resource pool is non-empty, and

means for determining if the acquiring user is at the head of the resource queue; and if the acquiring user is at the head of the resource queue:

means for removing the acquiring user from the resource queue;

means for removing at least one of the resources from the resource pool;

and

means for returning said at least one of the resources to the acquiring user,

otherwise, continuing to have the acquiring user wait for notification on the synchronization flag; and

means for the acquire resource means ~~to return a time out exception~~ remove the acquiring user from the resource queue if the acquiring user does not acquire at least one of the resources within a predefined time; and

synchronization means to constrain users with regard to a plurality of individual available resource objects such that only one user may execute either the release resource means or the acquire resource means at any one time, said synchronization means being synchronized on the synchronization flag.